

# Chap. 6 : Introduction aux images

## I. Généralités

Les images que vous voyez sur votre ordinateur sont stockées dans des fichiers. Il existe plusieurs formats d'images, c'est-à-dire plusieurs façons de coder l'image dans un fichier. Selon le format, l'image prend plus ou moins d'espace disque et est de plus ou moins bonne qualité.

### I.1. Image.bmp

- Le Bitmap (bmp) est un format non compressé (contrairement aux JPG, PNG, GIF etc.)

Concrètement, cela signifie les choses suivantes :

Le fichier est très rapide à lire, contrairement aux formats compressés qui doivent être décompressés.

La qualité de l'image est parfaite. Certains formats compressés comme le JPG détériorent la qualité de l'image, ce n'est pas le cas du BMP.

Mais le fichier est aussi bien plus gros à charger puisqu'il n'est pas compressé !



### I.2. Image.jpeg ou Image.jpg

- Le format jpeg ou jpg (Joint Photographic Experts Group) est un format de compression très efficace mais avec perte de qualité. Plus l'image est compressée, plus la qualité de l'image diminue. Il convient donc de trouver un compromis permettant un chargement rapide tout en gardant une qualité acceptable. Les logiciels graphiques permettent de régler le facteur de compression qui va de 1 à 99.

Le format JPEG est particulièrement adapté et recommandé pour les images de type Photographies, il peut supporter en effet 16.7 millions de couleurs et donne un bon rendu pour les images nuancées et les dégradés. Le format JPEG ne supporte pas la transparence et ne permet pas de créer des animations.

5466 octets

2506 octets

1147 octets



compression faible

compression moyenne

compression forte

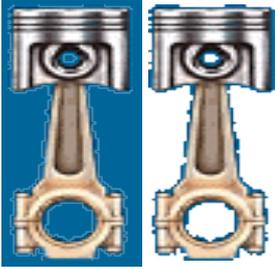
JPEG: La qualité de l'image diminue avec l'augmentation du facteur de compression

### I.3. Image.gif

Le format GIF (Graphics Interchange Format) est un format qui utilise une compression sans perte de qualité.

Les images au format GIF peuvent contenir un maximum de 256 couleurs, ce qui rend ce format peu adapté pour les photographies, les images nuancées et les dégradés. Par contre ce format donne d'excellents résultats pour les images avec un petit nombre de couleurs, les logos, les formes géométriques, boutons etc...

Le format GIF supporte la transparence et il permet également de créer des animations : les GIFs animés. L'option d'entrelacement permet d'afficher l'image progressivement, l'image étant flou au début du chargement et devenant de plus en plus précise.



Le même GIF animé transparent sur 2 fonds différents.

A droite le contour n'est pas net. Pour éviter cela, il peut être utile dans certains cas de créer les GIFs transparents (animés ou non) avec une couleur d'arrière plan qui se rapproche du fond sur lequel ils seront utilisés avant de rendre cette couleur d'arrière plan transparente.

2879 Octets



243 couleurs utilisées

2232 Octets



32 couleurs utilisées

901 Octets



16 couleurs utilisées

Incidence du nombre de couleurs sur le poids et l'apparence des GIFs

#### I.4. Image.png

Le format PNG (Portable Network Graphics) utilise une compression sans perte de données.

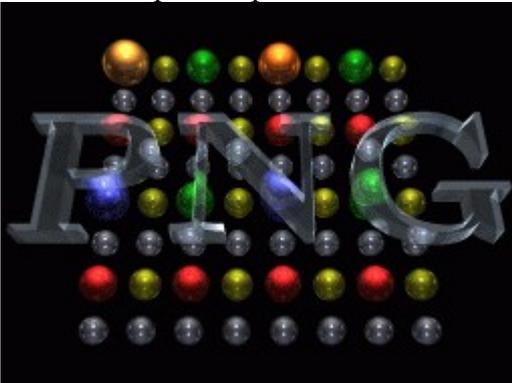
Le PNG supporte comme le GIF une palette indexée jusqu'à 256 couleurs, mais également les niveaux de gris jusqu'à 16 bits et les couleurs réelles jusqu'à 42 bits.

Le PNG supporte la transparence ainsi que 254 niveaux de transparence partielle.

Il possède également une méthode d'entrelacement très performante.

Les fichiers PNG sont généralement plus légers que les GIF, par contre le PNG est moins performant que le JPEG pour la compression des photographies.

Le PNG ne permet pas de créer des animations.



## II.Charger des donnée (*Load*)

### II.1.Structure

Pour charger des données (images, vidéo, ...) on utilise la class Loader.

Vous allez réaliser un application qui permet de charger une image sur la scène.

Pour cela, vous allez créer un nouveau projet AS3 appeler imageLoader.

Ensuite recopier le code ci-dessous pour placer l'image sur la scène.

```
public class Main extends Sprite
{

    public function Main():void
    {
```

```

        if (stage) init();
        else addEventListener(Event.ADDED_TO_STAGE, init);
    }

    private function init(e:Event = null):void
    {
        removeEventListener(Event.ADDED_TO_STAGE, init);
        // entry point
        // création du conteneur de l'image
        var conteneurImage:Loader = new Loader();
        // url de l'image à charger
        var adresseImage : URLRequest = new
            URLRequest("http://www.sciencesappliquees.com/terrain.jpg");

        // Événement pour voir si l'image est complètement charger
        conteneurImage.contentLoaderInfo.addEventListener(Event.COMPLETE,
            imageCharger);

        //lancement du chargement
        conteneurImage.load(adresseImage);
    }

    // Le chargement de l'image du terrain est terminé
    private function imageCharger(e:Event):void
    { //Sauve le chargement dans la variable bitmap locale image
        var image:Bitmap = (Bitmap) (e.target.content);
        //tester
        // met l'image sur la scene
        addChild(image);
        //tester
    }
}

```

Pour en savoir plus sur la class Loader :

[http://livedocs.adobe.com/flash/9.0\\_fr/ActionScriptLangRefV3/flash/display/Loader.html](http://livedocs.adobe.com/flash/9.0_fr/ActionScriptLangRefV3/flash/display/Loader.html)

## II.2.Programmation

Compléter le code ci-dessus pour redimensionner l'image à la scène.

Charger une nouvelle image sur la scène : "http://www.sciencesappliquees.com/ballon.png"

Faire bouger le ballon avec la souris.

Mettre en rotation le ballon à l'aide de la class Timer.